

100% Pass 2026 High Hit-Rate DEA-C02: Dumps SnowPro Advanced: Data Engineer (DEA-C02) Collection



2026 Latest iPassleader DEA-C02 PDF Dumps and DEA-C02 Exam Engine Free Share: <https://drive.google.com/open?id=18eQ7-2cfQIVZhgk7cgUFCqedIta7VLhp>

Obtaining the DEA-C02 certificate will make your colleagues and supervisors stand out for you, because it represents your professional skills. At the same time, it will also give you more opportunities for promotion and job-hopping. The DEA-C02 latest exam dumps have different classifications for different qualification examinations, which can enable students to choose their own learning mode for themselves according to the actual needs of users. On buses or subways, you can use fractional time to test your learning outcomes with DEA-C02 Test Torrent, which will greatly increase your pro forma efficiency.

There may be some other study materials with higher profile and lower price than our products, but we can assure you that the passing rate of our DEA-C02 learning materials is much higher than theirs. And this is the most important. According to previous data, 98 % to 99 % of the people who use our DEA-C02 Training Questions passed the exam successfully. If you are willing to give us a trust on our DEA-C02 exam questions, we will give you a success.

>> Dumps DEA-C02 Collection <<

Best DEA-C02 Preparation Materials, New Exam DEA-C02 Materials

Do you want to spend half of time and efforts to pass DEA-C02 certification exam? Then you can choose iPassleader. With efforts for years, the passing rate of DEA-C02 exam training, which is implemented by the iPassleader website worldwide, is the highest of all. With iPassleader website you can download DEA-C02 free demo and answers to know how high is the accuracy rate of DEA-C02 test certification training materials, and to determine your selection.

Snowflake SnowPro Advanced: Data Engineer (DEA-C02) Sample Questions (Q350-Q355):

NEW QUESTION # 350

A provider account is sharing a database named 'SHARED DB' through a share named 'MY SHARE'. The consumer account has created a database named 'CONSUMER DB' from the share. The provider account revokes access to a table named 'SALES DATA' within 'SHARED DB'. What will happen when a user in the consumer account attempts to query 'CONSUMER DB.SHARED SCHEMA.SALES DATA'?

- A. The query will fail with an error message indicating that the table does not exist or the user does not have privileges.
- B. The query will execute successfully, but only return data that existed before the access was revoked.
- C. The query will be cached based on the initial access, so users can continue query previous result based on same SQL.
- D. The query will be automatically re-routed to another available share containing 'SALES DATA'.
- E. The query will execute successfully, but the user will receive an empty result set.

Answer: A

Explanation:

When access to a shared object is revoked in the provider account, the consumer account loses access to that object. Subsequent queries in the consumer account will fail with an error message indicating insufficient privileges or that the object does not exist. Snowflake does not automatically redirect queries to other shares or provide cached result once access is revoked.

NEW QUESTION # 351

You have a data pipeline that loads data from an internal stage into a Snowflake table (Craw_data). The pipeline is experiencing intermittent failures with the error 'SQL compilation error: Stage 'MY INTERNAL STAGE' is immutable'. What are the potential causes of this error and how would you troubleshoot it?

- A. The internal stage is being used by multiple COPY INTO commands simultaneously, causing a resource contention issue. Implement queuing or throttling mechanisms to manage concurrent data loading.
- B. The user executing the COPY INTO command lacks the necessary privileges (USAGE on the stage). Grant the appropriate privileges to the user or role.
- C. This error is caused by insufficient warehouse size. Increase the warehouse size to accommodate the COPY INTO operation.
- D. The internal stage has been accidentally dropped and recreated with the same name during the COPY operation. Verify the stage's existence and creation timestamp.
- E. Another concurrent process is attempting to drop or alter the internal stage while the COPY INTO command is running. Implement proper locking mechanisms to prevent concurrent modifications.

Answer: D,E

Explanation:

The 'Stage is immutable' error typically indicates that the stage's definition has changed during the COPY operation. This can happen if the stage is dropped and recreated (option A) or if another process is altering the stage concurrently (option C). Privilege issues (option B) would usually result in a different error message. Resource contention (option D) is less likely to cause this specific error but could impact performance. Warehouse size (option E) is generally not directly related to this error.

NEW QUESTION # 352

A financial institution is using Snowflake to store transaction data for millions of customers. The data is stored in a table named 'TRANSACTIONS' with columns such as 'TRANSACTION ID', 'CUSTOMER ID', 'TRANSACTION DATE', 'TRANSACTION_AMOUNT', and 'MERCHANT CATEGORY'. Analysts are running complex analytical queries that often involve filtering transactions by 'TRANSACTION_DATE', 'MERCHANT_CATEGORY', and 'TRANSACTION_AMOUNT' ranges. These queries are experiencing performance bottlenecks. The data team wants to leverage query acceleration service to improve performance without significantly altering the existing query patterns. Which of the following actions or combination of actions would be MOST beneficial, considering the constraints and the nature of the queries? (Select TWO)

- A. Create materialized views pre-aggregating the transaction data by 'MERCHANT_CATEGORY' and 'TRANSACTION_DATE', and enable query acceleration on the virtual warehouse.
- B. Enable Search Optimization Service for the 'TRANSACTIONS' table, specifically targeting the 'MERCHANT_CATEGORY' column. Enable query acceleration on the virtual warehouse.
- C. Increase the size of the virtual warehouse used for running the queries and enable query acceleration on the warehouse without further modifications.
- D. Enable Automatic Clustering on the 'TRANSACTIONS' table, ordering the keys as 'TRANSACTION_DATE', 'MERCHANT_CATEGORY', 'CUSTOMER_ID'. Then, enable query acceleration on the virtual warehouse.
- E. Create separate virtual warehouses dedicated to reporting queries and ad-hoc queries respectively. Enable query acceleration only for the warehouse running reporting queries.

Answer: B,D

Explanation:

Enabling Automatic Clustering on 'TRANSACTIONS' with the specified key order ('TRANSACTION_DATE', 'MERCHANT_CATEGORY', 'CUSTOMER_ID') aligns the data layout with common query patterns, allowing Snowflake to efficiently prune irrelevant data during query execution. This drastically improves query performance. Enabling Search Optimization on the 'MERCHANT_CATEGORY' further enhances query performance by creating search access paths that enable faster lookups and filtering based on merchant category. Simply increasing the warehouse size (option A) may provide some improvement, but it's less targeted and potentially less cost-effective than optimizing the data organization. While dedicated warehouses (option C) can improve concurrency, they do not address the underlying performance bottleneck related to data access. Materialized views (option

E) can be beneficial, but they require careful design and maintenance, and they might not be flexible enough for ad-hoc queries with varying filter conditions. Clustering and search optimization provide a more general and efficient solution in this scenario.

NEW QUESTION # 353

You are designing a Snowpipe pipeline to ingest data from an AWS SQS queue. The queue contains notifications about new files arriving in an S3 bucket. However, due to network issues, some notifications are delayed, causing Snowpipe to potentially miss files. Which of the following strategies, when combined, will BEST address the problem of delayed notifications and ensure data completeness?

- A. Implement a Lambda function that triggers the 'SYSTEM\$PIPE FORCE RESUME' procedure after a certain delay.
- B. Configure the SQS queue with a longer retention period and implement an event bridge rule with a retry policy to resend notifications.
- C. Increase the 'MAX FILE AGE' parameter in the Snowpipe definition and implement a periodic 'ALTER PIPE ... REFRESH' command.
- D. Use 'VALIDATE()' function periodically to identify files that have not been loaded and trigger manual data loads for missing data.
- E. Set 'MAX FILE AGE' to 'DEFAULT' and utilize the 'SYSTEM\$PIPE FORCE RESUME' procedure in conjunction with a separate process that lists the S3 bucket and compares it to the files already loaded in Snowflake, loading any missing files.

Answer: E

Explanation:

Option E provides the most robust solution. Setting 'MAX FILE AGE' to 'DEFAULT' ensures that Snowpipe considers all files, regardless of their age. 'SYSTEM\$PIPE FORCE RESUME' can help in some cases. The key component is the supplemental process that actively compares the S3 bucket contents with the loaded data, identifying and loading any missing files due to delayed notifications. This approach guarantees data completeness even with delayed or missed SQS notifications. A,B,C,D doesn't guarantee data completeness. A alone can cause huge latency issue. B is not optimal. C addresses the SQS issue but not guaranteed to catch every case. D requires manual load intervention

NEW QUESTION # 354

You are responsible for optimizing query performance on a Snowflake table called 'WEB EVENTS, which contains clickstream data'. The table has the following structure: CREATE TABLE WEB_EVENTS (event_id VARCHAR(36), user_id INT, event_time TIMESTAMP NTZ, event_type VARCHAR(50), page_url VARCHAR(255), device_type VARCHAR(50) Users frequently run queries that filter the 'WEB_EVENTS' table based on a combination of 'event_type', and a date range derived from 'event_time' You observe that these queries are consistently slow Which of the following strategies would be MOST effective in improving the performance of these frequently executed queries?

- A. Create a clustering key on 'event_time' .
- B. Create a search optimization service on the 'page_url' column.
- C. Add a column to the 'WEB_EVENTS' table for the date part of 'event_time' and create a clustering key using the new date column along with and device_type' .
- D. Create a clustering key with the following order: 'event_type', 'device_type', 'event_time' .
- E. Create a materialized view that pre-aggregates data by 'event_type', 'device_type', and day (derived from 'event_time').

Answer: D,E

Explanation:

Options C and D are the most effective. A materialized view (C) would pre-compute the frequently requested aggregations, significantly reducing query execution time. Creating a clustering key on 'event_type', and 'event_time' (D) would optimize data organization for the common filter criteria, improving micro-partition pruning. Clustering only on 'event_time' (B) might help with date range filtering but won't address the 'event_type' and 'device_type' filters. Search optimization service (A) is designed for point lookups on string values and is not appropriate for this scenario. Adding a separate column for the date part of the event (E) is redundant and unnecessary. Snowflake can extract date parts from a timestamp column.

NEW QUESTION # 355

.....

2cfQIVZhkg7cgUFCqedlta7VLhp